EKXL: A Dynamic Poisson–Solver for Spherically–Convergent Inertial–Electrostatic Confinement Systems†

Katherine E. King and Robert W. Bussard

EMC2–1191–03

† This work performed under Contract No. MDA–972–90–C–0006 for the Defense Advanced Research Projects Agency, Defense Sciences Office.

ENERGY/MATTER CONVERSION CORPORATION
9100 A Center Street, Manassas, VA 22110, (703) 330-7990

EKXL: A DYNAMIC POISSON–SOLVER FOR SPHERICALLY–CONVERGENT INERTIAL–ELECTROSTATIC CONFINEMENT SYSTEMS

I. HISTORY AND BACKGROUND OF THE EKXL CODE

The EKXL code, and each of its predecessors, is a 1–D radially–dependent Poisson–solver. It gives static solutions for potential and density distributions, given edge density specifications for these parameters. It is a "point" model in that spatial variation of externally–imposed fields is possible, but spatial dependence of inter–particle interaction phenomena can not be accommodated as an integral part of code operations.

The original code, IACCEL2/EACCEL2, was developed by Ensley[1, 2] and provided closed analytic solutions in integral form using Struve and Bessel functions. This code employed Gaussian distributions for transverse energy and momentum. Solutions were limited, however, to electron currents of 1–10 A and core densities of $1x10^{9}$–$1x10^{10}$ $1/cm^{3}$, as indicated in Figure 1.

The first modification to Ensley's work was ACCEL, developed by Mission Research Corporation[3] (MRC). This version of the code utilized a more efficient algorithm for converging a solution to Poisson's equation, resulting in significant run–time savings. New physics elements were also added in order to include tangential current densities where previously only radial current was used. As shown in Figure 1, with the ACCEL improvements solutions were achieved with electron currents of 100–500 A and core densities of $1x10^{11}$–$1x10^{12}$ $1/cm^{3}$.

What followed was a new code, XL, also developed by MRC[4], that was utilized to its capacity by EMC[2] to study the Polywell^{tm}/HEPS system[5]. This code employed a

differential numerical analysis procedure and matrix inversion solution method. The distribution functions for transverse energy and momentum were changed to square functions to simplify the calculation. Features added at this time were adjustable precision and an adaptive grid. This code achieved much higher electron recirculating currents, $1x10^9$ A, and core densities, $1x10^{13}$–$1x10^{14}$ $1/cm^3$. However, the core density barrier at $1x10^{14}$ $1/cm^3$ could not be surpassed, making any studies of Polywell^{tm} reactor densities impossible. Work was then underway at MRC on the next revision, KXL.

II. THE ADIABATIC KXL CODE

KXL uses the same mathematical engine developed in the XL code and a series of intermediate "steady–state" solutions over time to reach the final steady–state condition for the given input parameters. Selected parameters, such as input current and the length of the time step, can be adjusted as the calculation proceeds. As indicated in Figure 1, this method and some of the improvements discussed below allow the code to reach much higher core densities than were possible with XL.

The "trapping" formula, F_{trap}, allows feedback on the current density to adjust the input current. The algorithm adjusts the current based on the core potential to prevent the anode from maximizing, due to too many ions in the core, causing the calculation to be unsuccessful. A target anode potential and maximum beam current are input and the code adjusts the beam current up to that maximum in an attempt to obtain the target anode height. The input variable is V_0 and it represents the target well depth. V_0 can be different for each species, allowing separate control over each beam. For example, a V_0 for the electron beam greater than the maximum well depth would set F_{trap} equal to one and allow the electron beam to remain on at maximum current for the entire run while the code adjusted the ion current within the indicated range. The anode control formula is:

$$F_{trap} = \frac{1}{1+e^{\alpha}} \qquad \text{where} \quad \alpha = \frac{\varphi_0 - V_0}{\pm .03 \ \varphi_{max}}, \tag{1}$$

φ_0 = Current Core Potential, φ_{max} = Maximum Core Potential, and the ± is determined by the charge on the beam.

If the maximum available ion current is too high, however, the F_{trap} formula cannot turn the current down enough. In effect, there is gun "leakage" and the ions build up (due to their recirculation) and overwhelm the core. When using a large ion current, it still needs to be within a reasonable range of the final gun current for the code to run successfully.

The adaptive grid allows for a specified percentage of grid points to be redistributed by the code, as needed, to resolve high density and/or high gradient regions. A radial weighting factor ("R" weighting) was first used to ensure an adequate number of grid points near the outer radius, a region where the density changes rapidly. However, this weighting of the outer region did not allow sufficient grid points in the core to resolve high density cases. The adaptive gridding was then changed to weight on density as opposed to radius. Once this was done, the cases began to run more consistently and the core resolution was greatly improved. Figure 2a shows a KXL run with the "R" weighting and Figure 2b shows the same case with the "n" weighting. Note the improved resolution of the core region in Figure 2b.

The KXL code was run on the Cray computer to reach the high density solutions that appeared unattainable on the PC, in an attempt to fill in high density points in a study of core density vs. electron current. Examining this data, however, it became apparent that the core densities were much lower than expected from earlier calculations. Numerous variations of input variables were tried in an attempt to increase the densities. In the mean time, an explanation was sought as to why the densities were lower than predicted,

and attempts were made to model this behavior. Also, the slope of core density vs. electron current was expected to be linear for constant electron confinement. As Figure 3 indicates, it was not linear until the electron current reaches 1000 A.

The explanation for the change in slope at 1000 A was found, but without illuminating the too–low density problem. The electron density is increasing linearly the entire time, as expected. It is only the ion density that changes, due to the anode height. The δn required to form an anode is approximately $\delta n = (n_i - n_e) = 3.32 \times 10^9 \, (E_{anode}/r_o^2)$, for n in $1/cm^3$, E in keV and r_o in cm. The electron currents below the linear slope corresponded to densities where δn was a significant portion of the core density. As core density moves above this density, δn becomes less significant, the effect disappears, and the slope finally becomes linear. By plotting this data for varying anode heights it is apparent that the curve approaches the linear condition as the anode height approaches zero (Figure 4).

In an effort to understand the too–low core densities, attention turned to the transit times. The code was modified to print the transit time at each step and this information was plotted, revealing that the code–calculated transit times were much too small to correspond to reality (Figure 5). Several changes were suggested and tried to improve the transit times, however it was then found that altering the transit time calculation had no effect on the final solution. This led to a suspicion that the dn/dt equation, where n is the edge density, that was used to link the intermediate steady–state ("snapshot") solutions might be incorrect. In the KXL code this equation (for electrons) had the form:

$$\frac{dn_e}{dt} = \frac{n_{inj}}{\tau_t} - \frac{n_e}{\tau_{conf}}, \qquad (2)$$

which gives the final steady–state relation $n_{inj} G_j = n_e$. The basic premise of this time–dependent coupling equation (2) is, in fact, false. The correct formulation is one that connects time–dependence of <u>total</u> electron populations in the system between successive static solutions from the KXL code. This equation is:

$$\frac{d\bar{n}}{dt} = \frac{4\pi R^2 n_{inj} v_{inj}}{4/3\ \pi R^3} - \frac{\bar{n}}{t_c} \qquad (3)$$

However, since the codes are all written to solve only one–dimensional boundary–value problems, it was necessary to recast this equation (3) in terms of edge (boundary) densities. And, other physics phenomena of importance to system operation also needed to be added to improve the modeling capability of the code.

Taking all of these factors together, EMC^2 then began development of the <u>E</u>KXL code, using the mathematical engine of KXL but now including all of the appropriate ion/electron/plasma physics analytic forms appropriate to and required for spherically–convergent inertial–electrostatic confinement systems.

III. THE EKXL CODE

Change in Edge Density with Respect to Time

The first step in customizing EKXL to the Polywell^{tm} system was to derive a new, correct dn/dt equation. This equation is the key to the linkage of intermediate steady–state solutions over time, and numerical tests have shown solutions to be very sensitive to errors in this equation. The change in the total number of particles in the system is simply the

input less the losses over a given period of time. Therefore, the change in the average number of particles with respect to time is given by:

$$\frac{d\bar{n}}{dt} = \frac{4\pi R^2 n_{inj} v_{inj}}{4/3\ \pi R^3} - \frac{\bar{n}}{t_c} = \frac{3n_{inj}}{R/v_{inj}} - \frac{\bar{n}}{t_c} \tag{3}$$

where t_c is the confinement time, $G_{i,j} t_{trans}$. The code is only a differencing solution of a one–dimensional boundary–value problem. Thus $d\bar{n}/dt$ can only be tied to the change in edge (boundary) density, as opposed to average density, so a parameter, $F_n = \bar{n}/n_e$, was introduced to relate edge density to average density. Since this ratio is not fixed, F_n is calculated in the code each time the dn/dt equation is solved (see below). A second parameter, the ratio of free–stream half–transit time with no potential well to the actual transit time,

$$F_t = \frac{R/v_{inj}}{t_{trans}}, \tag{4}$$

was introduced to allow the transit time to be factored out of the dn/dt equation. The final form of the dn/dt equation is then:

$$\frac{dn_e}{dt} = \left[\frac{3n_{inj}}{F_n F_t} - \frac{n_e}{G_{i,j}} \right] \frac{1}{t_{trans}} \tag{5}$$

Calculation of F_n

The average number of particles between each pair of grid points is used to calculate the number of particles in that shell. This is summed over the entire grid and divided by the

total volume to arrive at $\bar{n}$. A modification was necessary, however, for the electron calculation in the outer shell. In high density cases, the electron density change across the last grid space $\delta r = R - r_1 = R(1-<r_1>)$ can be very large (ca. 10^3–10^4x) and an algebraic average is not accurate. An exponential mean form was derived based on the assumption that the logarithmic gradient of the edge electron density varies inversely with the Debye length, λ_D, at the outer radius, R, $d(\ln n_r)/dr = -n_r/\lambda_R$, and with a parameter, β, (typically β similar to 0.7), that characterizes the drop in E between the outer radius and the first grid point, $<r_1>$. When the electron density begins to increase rapidly in this region, EKXL will switch from an algebraic average to the following:

$$\bar{n} = n_R \frac{2\sqrt{\beta}}{(1-<r_1>)} \frac{\lambda_D}{R} \tag{6}$$

Electron and Ion Recirculation

Although analytic forms have been derived for ion loss due to core upscattering[6, 7] and from fusion reactions[8], it has been found convenient to choose a constant ion recirculation, G_i, as an independent input to the code. Variation in G_i affects only the ion injection current, I_i, as the circulating ion current, $I_c = I_i G_i$, is fixed by the parameters of a given case. The code has the ability to adjust I_i as needed (see below).

The electron recirculation, G_j, may be input at a constant value or it may be calculated in the code using the physics of electron confinement under mirror reflection (MR) and wiffle ball (WB) modes. This model is discussed in detail in other EMC2 technical notes/reports[9, 10, 11] and its pertinent algorithms and defining equations are given in Figure 6. The magnitude of B is a separate input parameter for the mirror/wiffle model. The

auxiliary potential, introduced in KXL to model mirror reflection of the electrons, $\mu\nabla B = \frac{1}{2}mv_{\perp}^2$, in the cusp region, is input as the magnitude of $\mu\nabla B$.

Ion Losses Due to Fusion Reactions

Fusion reactions of ions in the core and surrounding region will consume ions. This source of loss is important only at conditions of high core density and large ion energy. The total fusion rate is calculated in the code as part of the fusion and gain calculations (see below). This allows for easy inclusion of the ion losses due to fusion in the ion dn/dt equation. If the total fusion rate is DD_{fus} fusions/sec, then the number of ions lost to fusion is $2DD_{fus}$ ions/sec. This loss term would go into the original dN/dt equation and the resulting dn_e/dt equation would be:

$$\frac{dn_e}{dt} = \left[\frac{3n_{inj}}{F_n F_t} - \left[\frac{n_e}{G_i} + \frac{DD_{fus}}{F_n F_{DD}} \right] \right] \frac{1}{t_{trans}} \tag{7}$$

where $F_{DD} = 4/3\ \pi R^3/t_{trans}$ is introduced to allow the transit time to be factored out.

Ion Losses Due to Core Upscattering

Ion losses due to core collisional upscattering can be accounted for by adding a loss term to the dn/dt equation for ions. Based on the work of Rosenberg and Krall[6], as modified by Bussard[7], the upscattering time for ions, t_{up}, is calculated each time the dn/dt equation is solved, from

$$\frac{1}{t_{up}} = \frac{Z^4}{1.892\text{x}10^{13}\ f^2(1+f)^2\ A^{1/2}} \frac{1}{R} \int \frac{n(r)}{E(r)^{3/2}}\, dr \tag{8}$$

A factoring parameter, $F_u = t_{up}/t_{trans}$, was introduced into the dn/dt equation so that the ion losses due to upscattering are given by $n_e/(F_u t_{trans})$. This form is easily incorporated into the ion dn/dt equation which becomes

$$\frac{dn_e}{dt} = \left[\frac{3n_{inj}}{F_n F_t} - n_e \left[\frac{1}{G_i} + \frac{1}{F_u} \right] - \frac{DD_{fus}}{F_n F_{DD}} \right] \frac{1}{t_{trans}} \qquad (9)$$

The fractional velocity spread, $f = \delta v/v$, is a specified input parameter. If $f = 0$, no ion upscattering is calculated. This parameter can be estimated by comparison of scattering collision rates in the core, edge, and mantle regions[12, 13, 14] of the system, comparing transverse momentum vs. radial momentum exchange processes. Such estimates are not yet complete; when available the analytic forms that describe them can be added to the dn/dt equation, as above.

Average Transit Time Calculations

The data in the code is quite sufficient to calculate the ion transit time directly. Using the potential at each grid point and the input parameters E_{inj}, dE and dE_{perp}, the average transit time across each grid point can be calculated and summed across the entire grid to arrive at the average ion transit time through the system.

The calculation of electron transit time, however, presents an interesting problem. The direct method used for ion transit time does not lend itself to an accurate calculation of the average electron transit time as the individual times vary greatly with the energy spread of the electrons. Electrons with a maximum dE_{perp} will be turned early and have short transit times while the electrons with maximum energy, E_{inj}+dE, and no dE_{perp} will pass through the bottom of the well and reach the core yielding much longer transit times.

Due to the dependence of F_n on transit time, a small deviation in average transit time could continue to feed back into a large error in dn/dt. In short, the detailed computational capability of the code would lead to amplification of numerical errors in original data. In this circumstance, the internal data calculation was abandoned in favor of an analytic form based upon system parameters, which yields a more accurate and consistent averaged result and utilizes less computational time. As derived and discussed in another EMC2 note[14] (see for definitions of terms) the electron transit time is:

$$t_{tot} = \frac{2R}{v_{in}} \left[\frac{<r_0> + \left[\frac{<r_w>^2 - <r_0>^2}{2<r_0>} \right]}{\left[\eta' + \frac{1}{2}(1-\alpha_q) \right]^{1/2}} + \frac{2}{\left[1 + \frac{1}{2}(1-\alpha_q) \right]^{1/2}} \left[\frac{1}{<r_w>^{1/2}} - 1 \right] \right] \qquad (10)$$

where $\alpha_q = \varphi_o/\varphi_{min}$, $\eta' = 1-\alpha_q(1-\eta_o)$, $\eta_o = E_{ao}/E_o$,

$<r_w> = <r_0> + \left[\frac{2}{m} <r_0>^2 \eta' \right]^{1/(m+2)}$ and $<r_0>$ is the core radius.

This is a calculation of the maximum transit time. But the electrons occupy a spread in total and transverse energy. Across this energy distribution the individual transit times can vary markedly. Analysis of this variation suggests an exponential distribution of transit times across the band. Integrated averaging over this variation yields a correction factor to give average transit times of the average electron in terms of the total maximum transit time of equation (10). This factor is found to be:

$$\bar{t}_{trans} = \frac{\ln \left[\frac{t_{trans}}{2R/v_{in}} \right]}{\left[\frac{t_{trans}}{2R/v_{in}} - 1 \right]} t_{tot} \qquad (11)$$

Fusion Rates and Power Gain

The KXL code contained a subroutine to calculate the approximate neutron rate (1/2 the fusion rate) for DD fusion. This subroutine was modified in EKXL to include fusion rate calculations for DT, D^3He and $p^{11}B$ based upon the deuteron density (as the code does not currently accommodate different electron and ion injection radii, necessary for center–of–momentum, center–of–geometry placement of these ions). Calculation of system gross gain for all four fuel combinations was also added to the EKXL code, where gross gain is defined as

$$G_{gr} = \frac{P_{fus}}{P_{inj}} = \frac{b_{ij} E_{fus} \int_0^R n_i^2(r)\,\sigma(E) v_i(r) 4\pi r^2 dr}{P_{e_{inj}} + P_{i_{inj}}} \qquad (12)$$

The densities used for each fuel combination were:

DT For DT fusion rates, the existing deuteron density, D, is divided in half and treated as 50:50 D:T

D^3He For a 50:50 D:^{3}He distribution the total D + ^{3}He density will be 2/3 the existing D density for the same potential distribution.

$p^{11}B$ For 50:50 p:^{11}B the total density of p + ^{11}B is 1/3 the existing D density.

Bremmstrahlung Losses

Bremmstrahlung will be an important energy loss mechanism from the system at high density conditions. This was estimated from standard formulae[16] for bremmstrahlung

radiation density, integrated over the system volume to give total bremmstrahlung output, ignoring self–absorption by the plasma and any internal reflection from the boundary walls. This is discussed in a separate EMC2 technical note[17]. The resulting expression is used in the EKXL code to calculate bremmstrahlung gross output as

$$P_{bremm} = 1.69x10^{-32} \int_0^R n_i(r) n_e(r) Z^2 \sqrt{E_e(r)} 4\pi r^2 dr \qquad (13)$$

This is added to the injection power to determine net power input and system net gain.

$$G_{net} = \frac{P_{fus}}{P_{inj} + P_{bremm}} \qquad (14)$$

Ion Current Adjustment

The problem with the anode control, discussed previously – allowing too many ions to build up in the system and thus requiring the maximum ion current to be within range of the final ion current – was addressed in EKXL with a different form of anode control that calculates the ion current density based on the core densities, the electron current density and $G_{i,j}$. The difference in core densities is given by[18]

$$n_i - n_e = 3.32 \times 10^8 \frac{(\varphi_o - \varphi_{min})}{r_0^2} \qquad (15)$$

where φ_o is the core potential and φ_{min} is the potential at the bottom of the well (both in eV), r_0 is the core radius in meters and density is in $1/m^3$. The difference in densities at the target anode potential, φ_a, is then

$$n_i' - n_e' = 3.32 \times 10^8 \frac{(\varphi_a - \varphi_{min})}{r_0^2} \tag{16}$$

The change in density from n_i to n_i' is then

$$\Delta n_i = n_i' - n_i = 3.32 \times 10^8 \frac{(\varphi_a - \varphi_{min})}{r_0^2} + (\Delta n_e + n_e) - n_i$$

or

$$\Delta n_i = 3.32 \times 10^8 \frac{(\varphi_a - \varphi_{min})}{r_0^2} - (n_i - n_e) + \Delta n_e \tag{17}$$

Substituting equation (15) into equation (17) gives

$$\Delta n_i = 3.32 \times 10^8 \frac{(\varphi_a - \varphi_o)}{r_0^2} + \Delta n_e \tag{18}$$

and

$$J_i = \frac{3.32 \times 10^8}{G_i r_0^2 \Delta t} (\varphi_a - \varphi_o) + J_e \frac{G_j}{G_i} \tag{19}$$

where $J_{i,e}$ are the circulating current densities.

Automatic Operating Mode

In the XL code a case achieved a steady–state solution or none at all. The KXL code has the ability to stop at any "snapshot" point by specifying the time duration in the input file. This allows a succeeding run to be started at the end of a previous calculation, using a special data file created when that calculation was completed, and to continue for the time duration specified in the new input file. This allows for the adjustment of time steps and

ion current necessary to achieve the final steady–state solutions previously unattainable with the XL code. The drawback is that the manual adjustments of duration time and ion currents require continuing operator input. In the EKXL code an "automatic" mode was introduced that eliminates the need for most manual restarts. In automatic mode, the code will run to the end of the specified time duration or a steady–state solution, whichever comes first, using a series of intermediate time steps (simulating the manual restarts required in KXL) automatically adjusting the length of each intermediate time step based upon whether or not the previous time step was successful. This is facilitated by the ion current control and allows the code to run unattended as far as is possible without requiring operator intervention.

IV. SAMPLE CODE RUN FOR EKXL V4.1

Figure 7 shows a sample input file for EKXL version 4.1 as it is described above. This calculation ran in automatic mode with fixed ion current for a 15 keV well in a 92 cm device. The name of the log file, a sample of which is shown in Figures 8 a–b, and the dataset file, a sample of which is shown in Figure 9, are taken from the name of the input file, with the extension indicating the nature of the data. For example, the input file in Figure 7 is named BREM2b.INP. The log file in Figure 8 is named BREM2b.LOG and the dataset file in Figure 9 is named BREM2b.OUT. The last character of the file name is used to indicate the number of the run. In this example, the "b" indicates that it is the second run and the "Initialization file" is set to "YES" in the input file. This calculation started where BREM2a.INP ended, and this is verified in the beginning of the log file (Figure 8 a) where it states that initialization file BREM2.TMP was read.

The log file prints run diagnostics of the numerical calculation[4] to help the user in determining if and when problems occurred in the calculation. Figure 8a shows the

beginning of the log file, while Figure 8b shows the end. As seen in the figures, the log file also contains the transit times (Tau), the electron confinement time, the ion upscattering time (if this calculation is being used), the radius of the wiffle ball, $<r_b>$, the actual ion and electron currents being used in the code, the fusion rates for DT, D^3He and $p^{11}B$, the plasma gain for all four fuels and the bremmstrahlung power (in watts) as calculated from the DD density. The end of the file indicates that the snapshot was saved and lists the files to which output was written.

Figure 9 shows only the first portion of an output dataset file. This file contains the potential and ion and electron densities at each grid point for every successful snapshot in the run. The output is formatted and comma–separated which allows for easy import into spreadsheet or graphics software. A sample plot of this output data is shown in Figure 10. The beam profiles from the input file have been printed across the top of this chart and the fusion rates have been imported from the log file.

Hundreds of cases have been run with the EKXL code modelling many areas of interest from basic experiments, to current SCIF parameters, to reactor regimes. Results of these code runs have been used extensively to better understand the physics involved in Polywell^{tm} devices and also to improve the modelling, especially as it is applied in the code. A compilation of results of specific interest to the SCIF experiment is presented in some detail in another EMC2 technical report on these computer simulations[19]. The potential of this code, however, has not been fully realized and there still remain many areas of interest that have not been explored.

LIST OF FIGURES

REFERENCES

[1] Ensley, D.L., "Physics Modeling Project", internal memorandum to Pacific–Sierra Research Corporation, October, 1986

[2] Ensley, D.L., "IACCEL2/EACCEL2 User Guide", internal memorandum to Pacific–Sierra Research Corporation, April 1987

[3] McDonald, J., B. Goplen, "ACCEL User's Manual", Mission Research Corporation Report MRC/WDC–R–180, November 1988

[4] Smithe, D., "XL Reference Manual", Mission Research Corporation Report MRC/WDC–R–226, January 1990

[5] King, K.E., R.W. Bussard, "Examination of Electrostatic Well Formation for Spherically Convergent Electron and Deuteron Beams", Energy/Matter Conversion Corporation Report EMC2–0990–02, September 1990 as presented at the American Physical Society Thirty–Second Annual Meeting of the Division of Plasma Physics, Cincinnati, OH, November 1990

[6] Rosenberg, M.(EMC2), N.A. Krall, "Ion Loss by Collisional Upscattering", Krall Associates Report KA–90–39, October 1990

[7] Bussard, R.W., "Core Collisional Ion Upscattering and Loss Time", Energy/Matter Conversion Corporation Report EMC2–1090–03, October 1990

[8] Bussard, R.W., "Fusion Lifetime Limits on Ion Upscattering", Energy/Matter Conversion Corporation Report EMC2–0291–04, February 1991

[9] Bussard, R.W., K.E. King, "Electron Recirculation in Electrostatic Multicusp Systems: I – Confinement and Losses in Simple Power Law Wells", Energy/Matter Conversion Corporation Report EMC2–0491–03, April 1991

[10] Bussard, R.W., K.E. King, "Electron Recirculation in Electrostatic Multicusp Systems: II – System Performance Scaling of One–Dimensional 'Rollover' Wells", Energy/Matter Conversion Corporation Report EMC2–0791–04, July 1991

[11] Bussard, R.W., K.E. King, "Phenomenological Modelling of Polywell^{tm}/SCIF Multi–Cusp Inertial–Electrostatic Confinement Systems", Energy/Matter Conversion Corporation Report EMC2–1191–02, November 1991 as presented at the American Physical Society Thirty–Third Annual Meeting of the Division of Plasma Physics, Tampa, FL, November 1991

[12] Rosenberg, M.(EMC2), N.A. Krall, "Ion Loss by Collisions Outside the Core", Krall Associates Report KA–91–02, January 1991

[13] Krall, N.A., "Anisotropy Profiles for the Ion and Electron Plasma in the Polywell^{tm}/SCIF", Krall Associates Report KA–91–09, April 1991

[14] Krall, N.A., "Core and Mantle Properties of Weibel Instability in the Polywell^{tm}/SCIF", Krall Associates Report KA–91–10, May 1991

[15] Bussard, R.W., K.E. King, "Electron Transit Time in Central Virtual Anode Wells", Energy/Matter Conversion Corporation Report EMC2–0291–03, February 1991

[16] Book, D.L., "NRL Plasma Formulary", 1987 revised, Naval Research Laboratory, Washington, D.C.

[17] Bussard, R.W., K.E. King, "Bremmstrahlung Radiation Losses in Polywell^{tm} Systems", Energy/Matter Conversion Corporation Report EMC2–0891–04, August 1991

[18] Bussard, R.W., G.P. Jellison, G.E. McClellan, "Preliminary Research Studies of a New Method for

Control of Charged Particle Interactions", Pacific–Sierra Research Corp. Report PSR 1899, 30 November 1988, Final Report under Contract DNA001–87–C–0052, Appendix A, Sect. on Negative Potential Well Formation

[19] Bussard, R.W., K.E. King, "Computer Modelling of SCIF Experiment Performance", Energy/Matter Conversion Corporation Report EMC2–0691–02, June 1991

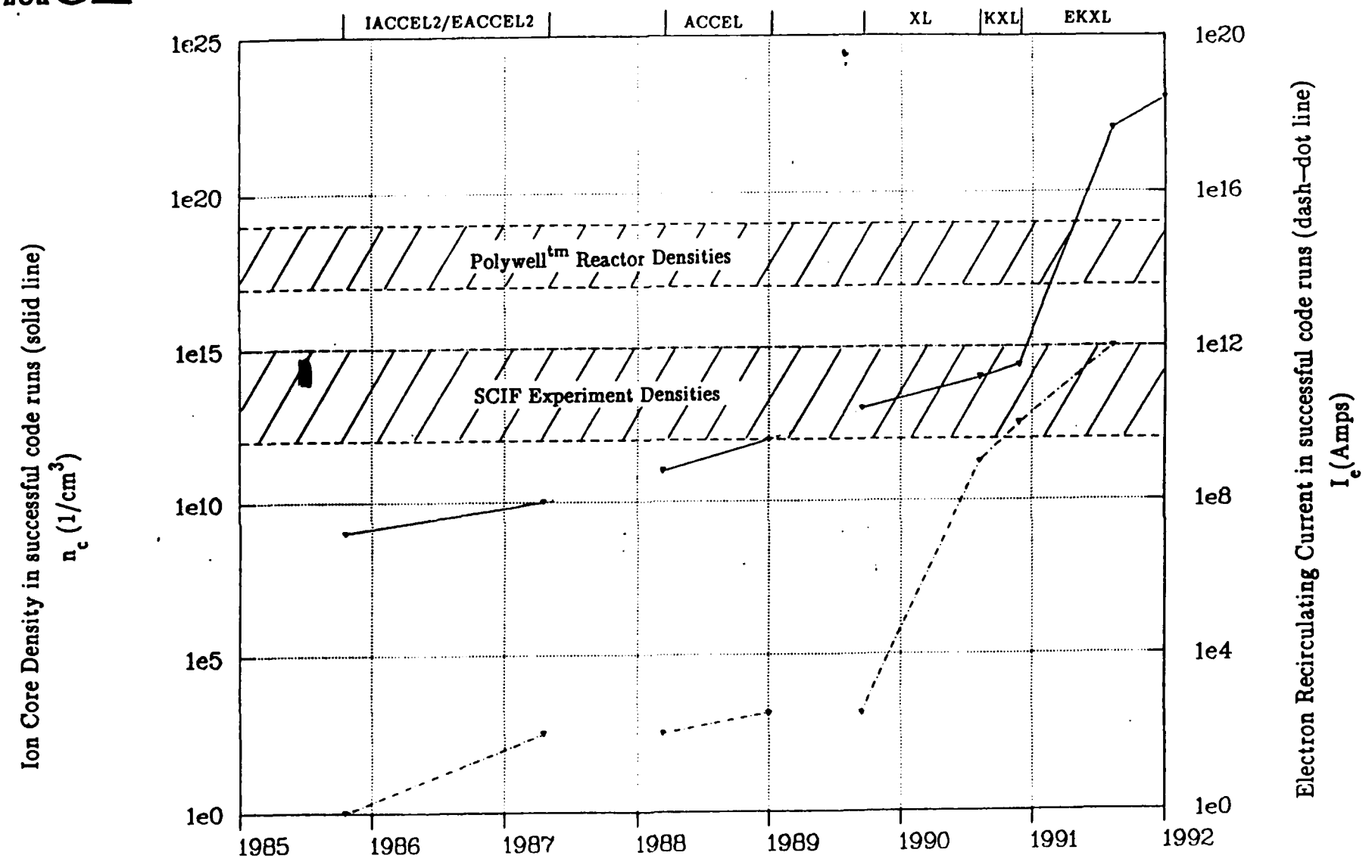

Figure 1.

Maximum Operational Levels
for Successful Vlasov–Poisson "1–D" Code Runs

Figure 2. a

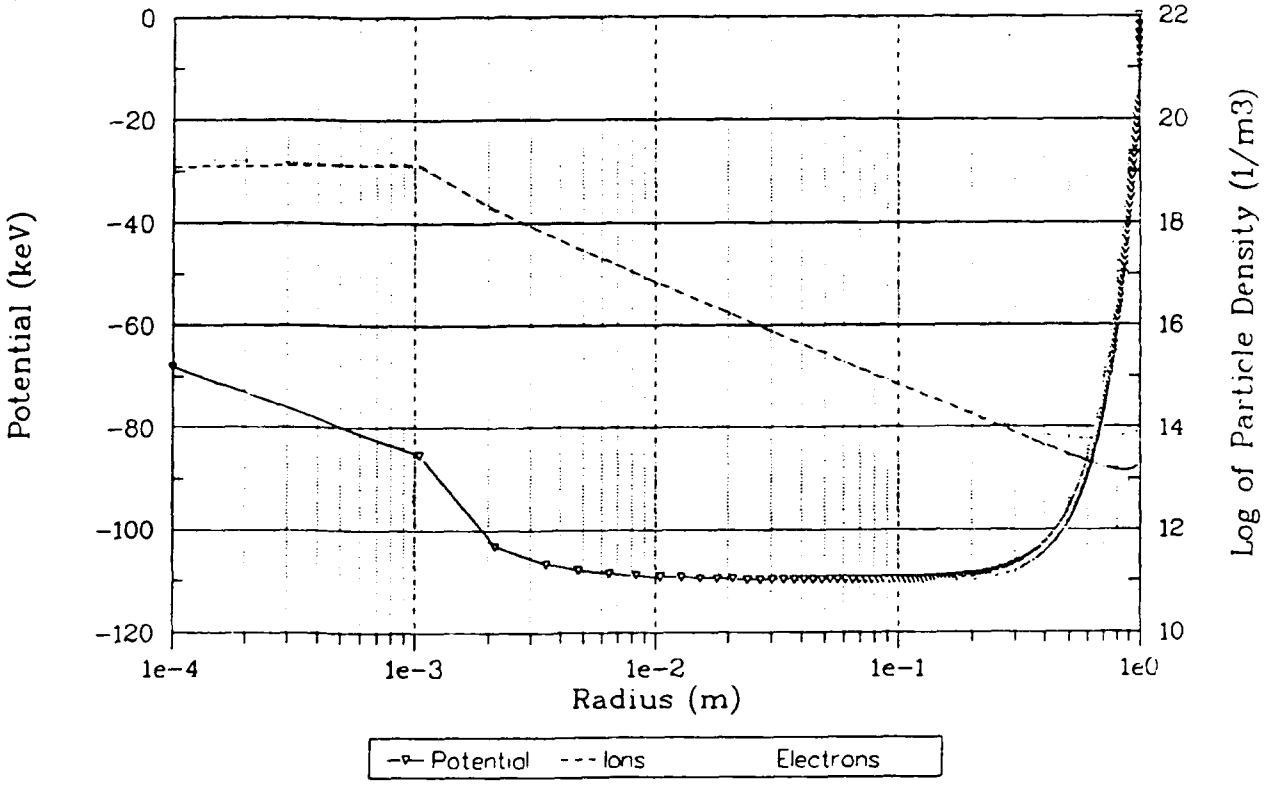

Figure 2. b

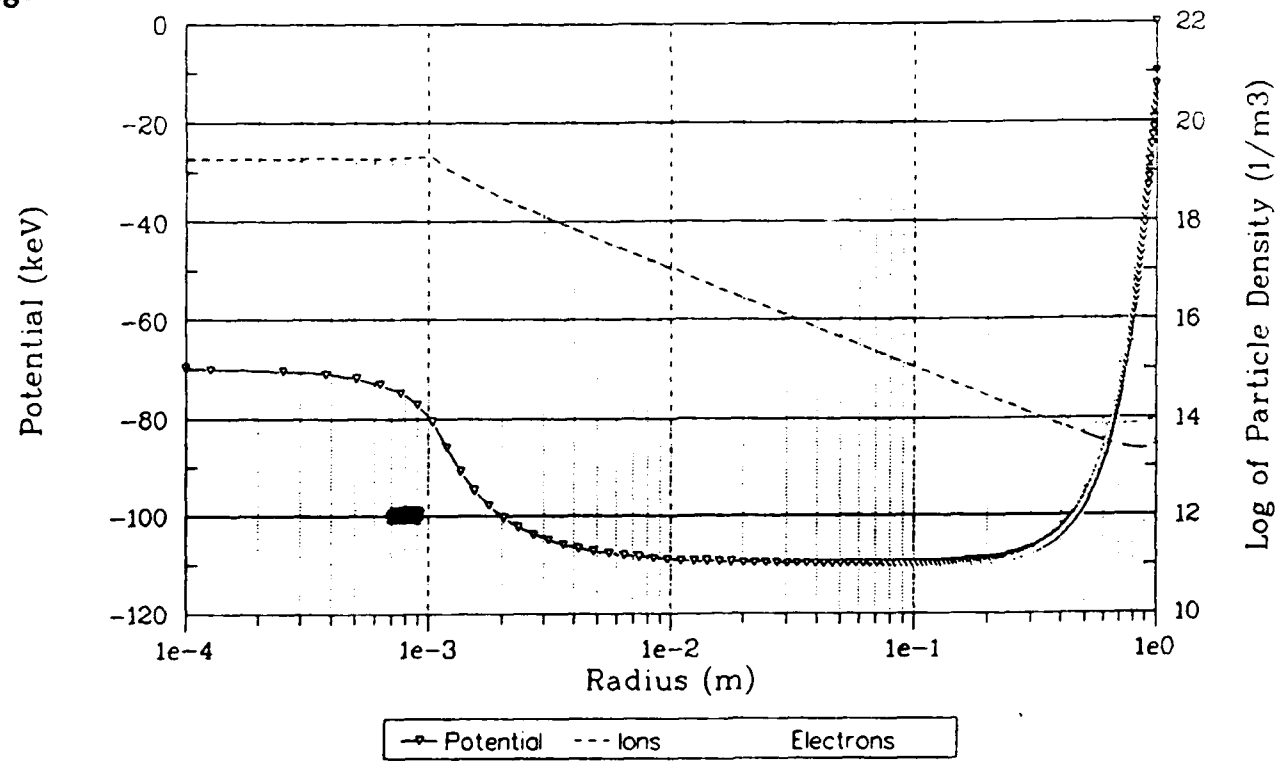

Figure 2. **a.** KXL case with R weighting factor.
b. Same case with n weighting. Note improved core resolution and density.

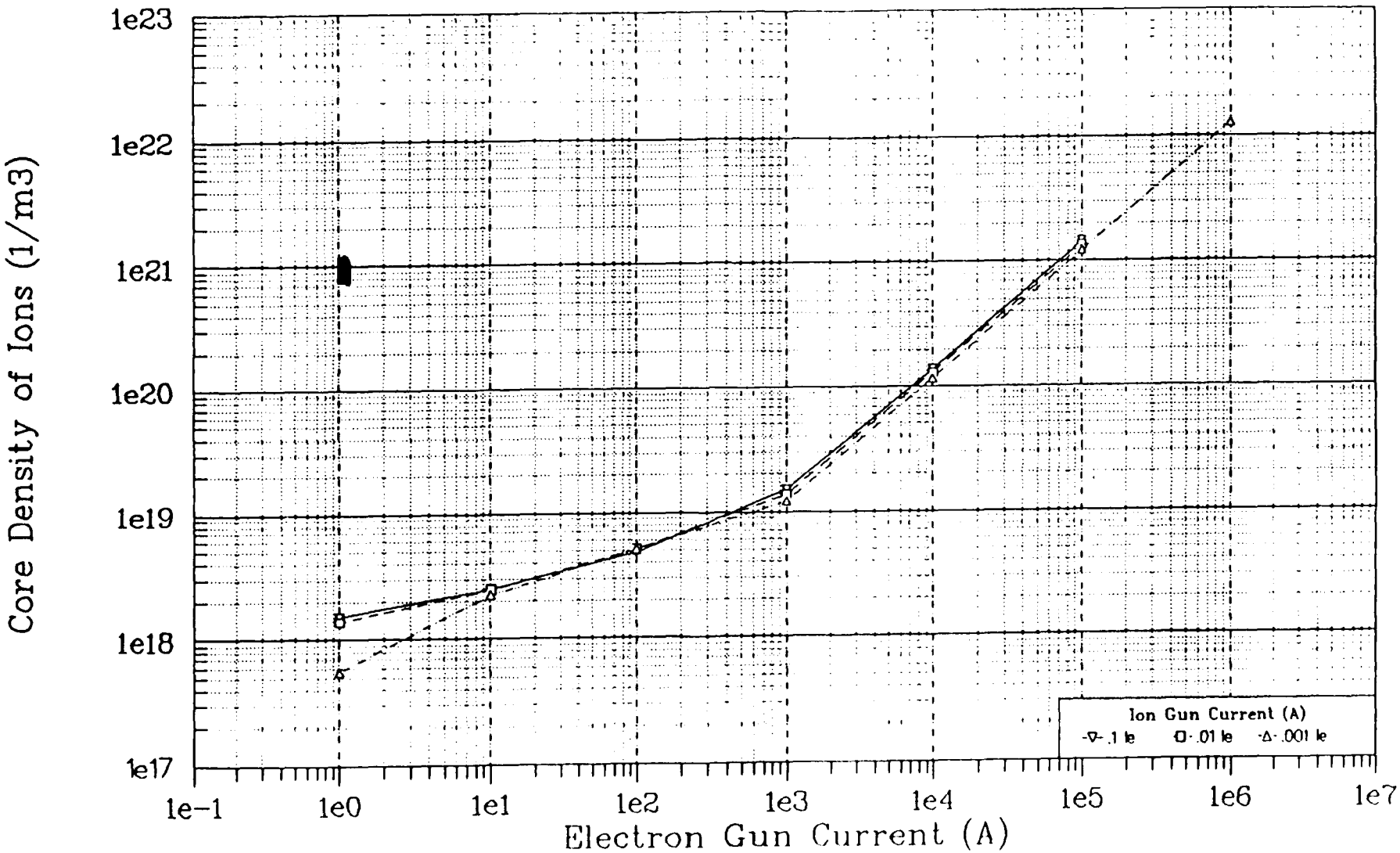

Figure 3. Core density vs. electron current for a fixed electron confinement. Slope is linear for currents greater 1000 A.

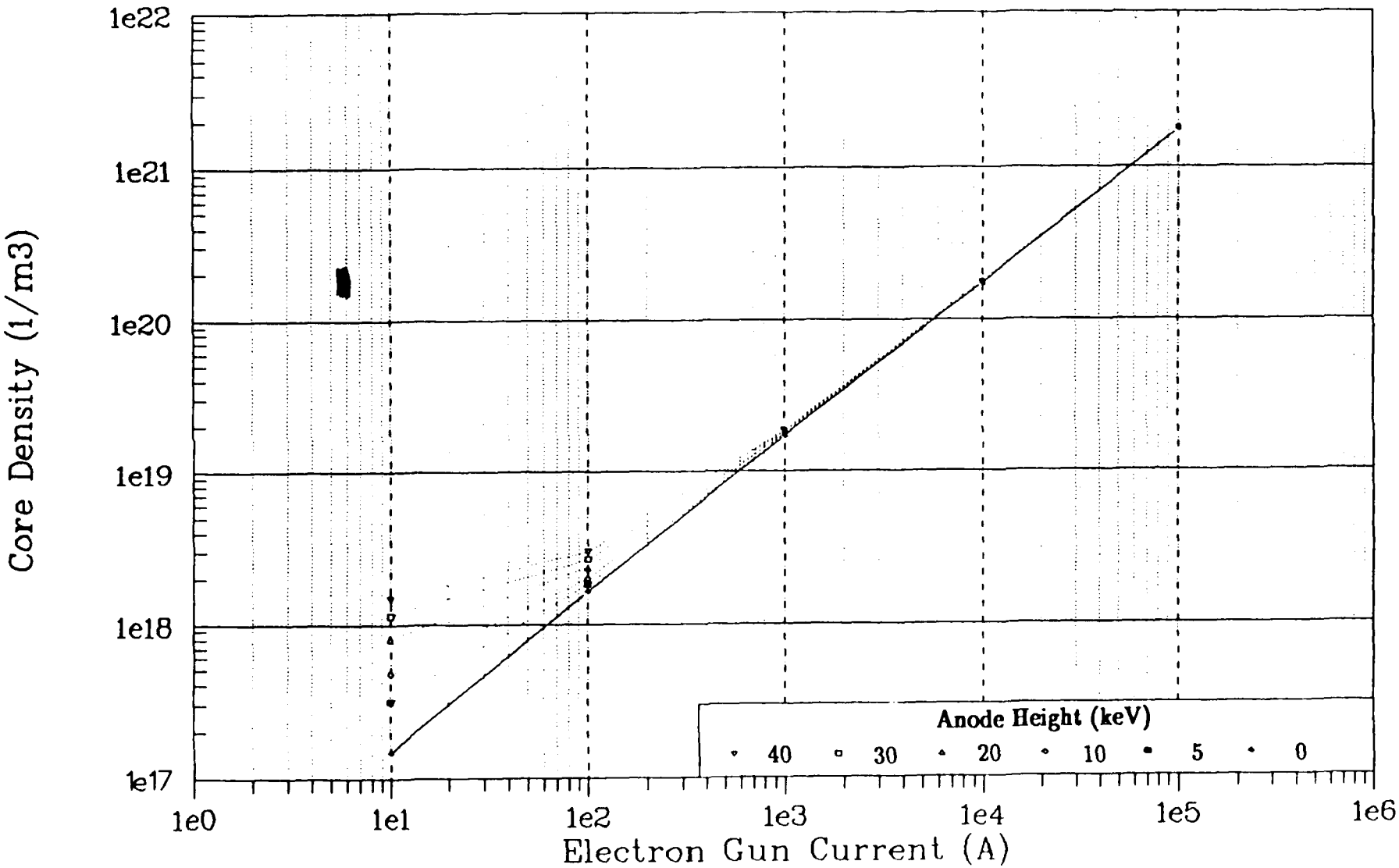

Figure 4. Core density vs. electron current for varying anode heights. Slope approaches linear as anode approaches zero.

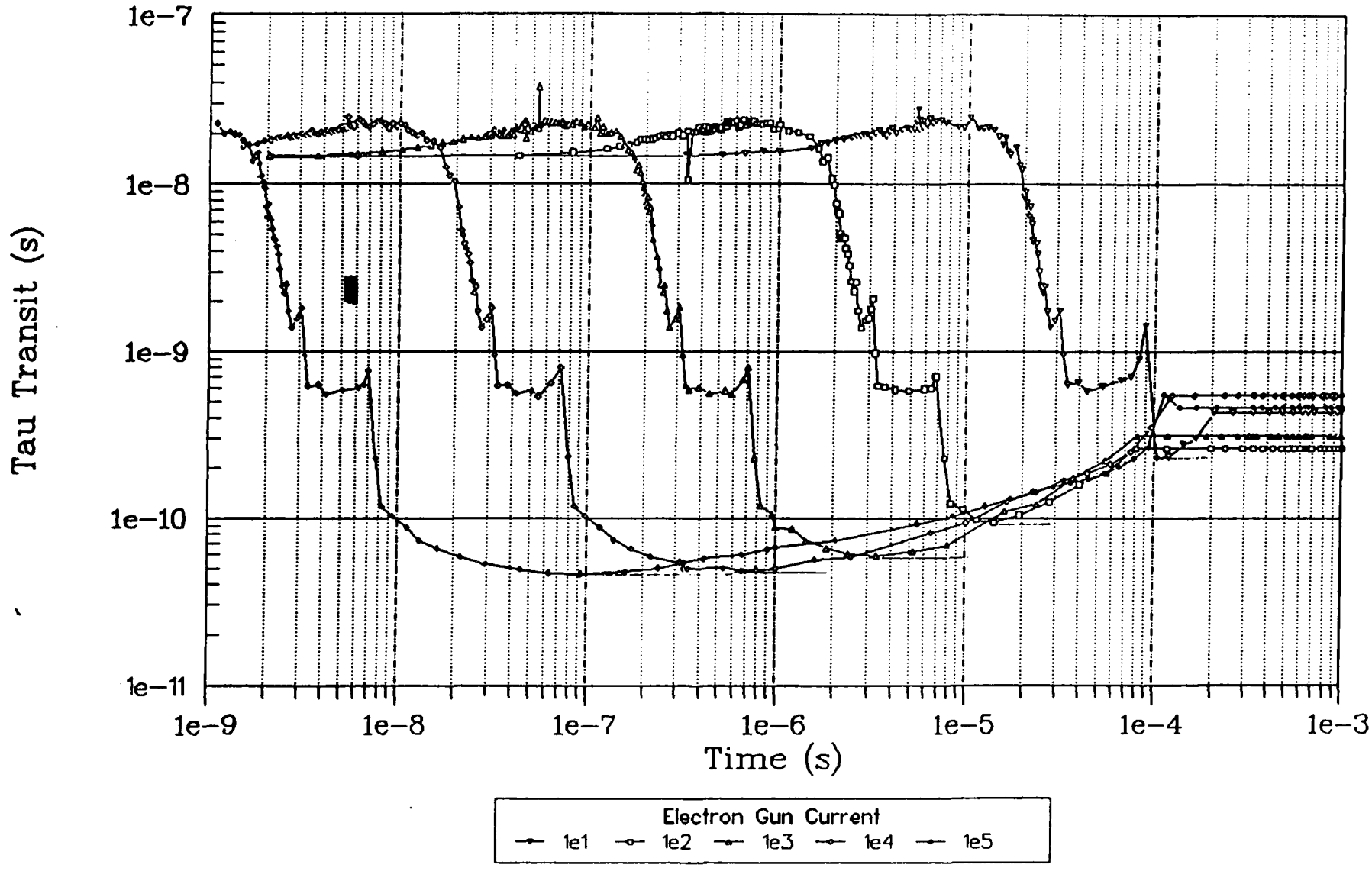

Figure 5. Erroneous electron transit time as calculated in KXL. For all currents, electron energy is 100 keV.

Mirror/Wiffle Model for G_{jo}

$$G_{jo} = G_{jMRo}\left[\frac{4}{N}\frac{(1-\langle r_b\rangle)}{(1-\alpha_r)}t_{MR} + \langle r_b\rangle t_{WB}\right]\frac{1}{t_{tot}} \qquad r_b < r_x$$

$$G_{jo} = G_{jMRo}\left[\frac{4}{N}\frac{(1-\langle r_b\rangle)}{(1-\alpha_r)}t_{MR} + G_{jWB}\langle r_b\rangle t_{WB}\right]\frac{1}{t_{tot}} \qquad r_b > r_x$$

where:

$$G_{jMRo} = \frac{1}{\langle r_{ad}\rangle^m}\left[1-\alpha_q(1-\langle r_{ad}\rangle^m)\right] \qquad r_b < r_{ad}$$

$$G_{jMRo} = \frac{1}{\langle r_b\rangle^m}\left[1-\alpha_q(1-\langle r_b\rangle^m)\right] \qquad r_b > r_{ad}$$

$$G_{jWB} = \frac{2r_e Z}{Nk_L^2 S^2} \qquad Z = 8\pi n_c r_c^2 \quad \text{and} \quad W = \frac{E_0}{B_0^2 R^2}$$

$$\langle r_b\rangle^{(2m+2)} = (ZW)\frac{\left[1-\alpha_q(1-\langle r_b\rangle^m)\right]}{\left[f_0(\langle r_b\rangle)\right]^2} \qquad \langle r_g\rangle^2 = \left[\frac{2WS^2}{r_e\langle r\rangle^{2m}}\right]\frac{\left[1-\alpha_q(1-\langle r\rangle^m)\right]}{\left[f_0(\langle r\rangle)\right]^2}$$

$$\langle r_x\rangle^{(2m+2)} = \left[\frac{NWk_L^2S^2}{2r_e}\right]\frac{\left[1-\alpha_q(1-\langle r_x\rangle^m)\right]}{\left[f_0(\langle r_x\rangle)\right]^2} \qquad S^2 = \overline{\sin^2\varphi}$$

$$\langle r_{ad}\rangle = \langle r_{ad0}\rangle + k_a(\langle r_b\rangle - \langle r_x\rangle) \qquad k_a = \frac{\langle r_k\rangle - \langle r_{ad0}\rangle}{\langle r_k\rangle - \langle r_x\rangle}$$

$$\langle r_{ad0}\rangle^{(2m+2)} = \left[\frac{2m^2}{\Gamma_a^2}\frac{WS^2}{r_e}\right]\left[1-\alpha_q(1-\langle r_{ad0}\rangle^m)\right]\left[1+\frac{\langle r_{ad0}\rangle}{m}\frac{f_0'(\langle r_{ad0}\rangle)}{f_0(\langle r_{ad0}\rangle)}\right]$$

$$f_0(\langle r\rangle) = \left[\frac{2}{1+\langle r\rangle^{(m+2)}}\right] \qquad f_0'(\langle r\rangle) = -\frac{2(m+2)\langle r\rangle^{(m+1)}}{\left[1+\langle r\rangle^{(m+2)}\right]^2}$$

Figure 6. Formulas used in the mirror/wiffle model for G_j calculation.

```
Log Filename:  (program uses INP filename)
---------------+=========================+
Dataset Name:  (program uses INP filename), Initialization file: YES,
---------------+=========================+----------------------[=]-
Operating mode:  AUTOMATIC   , end time=1.000e-03 sec, with   3 snapshots/dt,
-----------------+==========+-----------[=======]----------[==]---
Fixed Outer Potential:       0.e3 Volts at r=0.92   m, Ncusps=    8MR,   8WB,
------------------------+=======+-----------+====+-----------[==]---[==]---
Number of Beam Populations:   2, with repeller effic. :  0.9000,
-----------!--------------+==+--------------- -------!--+====+-
    Name     Ig [A]     Einj[eV]     dE[eV] dEperp[eV]  B[G],Gi     V0trap[eV]
1)  Elec     18.0e03     14.5e03     0.50e03     4.35e03     3.00e03    -20.0e03
2)  Deut     18.0e03    10.00e00     1.00e00     1.71e00        1e00    -11.0e03
----+==+--+=======+--+=======+--+=======+--+=======+--[=======]--[=======]
Number of Thermal Populations:   0,
--+~-----------------------------+==+-
    Name    [#/m**3]   @[Volts]   Temp[eV]
0)  Elec       1.e16      100.e3       20.e3
----+==+--+=======+--+=======+--+=======+
Deuterium Neutral Density:       0.e18 /m**3,  aij:           1 ICC mass ratio,
-----------------------------+=======+--------------+=======+---------------
Auxiliary Potential:      14.9e3 Volts for e's, with exponent=3.     ,
----------------------+=======+-------------------------------+====+-
Fractional velocity spread:      0, kl Radius Factor:     2.0,
-----------------------------[==]-------------------[====]-
Grid:  200 points with adaptive fraction of 0.8     and tolerance 1.e-5     ,
------+==+-----------------------------------+====+--------------+=======+-
Transit Time Correction Factor (Y/N): Y,  ka <radius>:   0.83.
---------------------------------------=--------------[====]-
Modified B Field Shape: Y, Modified Gj: Y, Adjust Ion Current: N.
-----------------------=--------------=---------------------=
Bremmstrahlung <radius>:   1.00,  Ion Fusion Losses:  N
-------------------------+===+----------------------=
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Figure 7. Sample input file for an EKXL v4.1 code run.

Figure 8. a

```
Operating mode is AUTOMATIC

Auxiliary potential option is on.

Initialization file read:  brem2.tmp

iter =  1, dumax = 3.21E-07, step = 1.00E+00
     Electron confinement time:  6.389E-07
    Wiffle radius rb:  1.936E-01
    1)   time = 1.000E-06, determ = 1.000E+20 Tau = 6.100E-08 3.019E-06
    2)   time = 1.263E-06, determ = 1.000E+20 Tau = 6.100E-08 3.027E-06
    3)   time = 1.395E-06, determ = 1.000E+20 Tau = 6.100E-08 3.019E-06
    4)   time = 1.972E-06, determ = 1.000E+20 Tau = 6.100E-08 3.022E-06
    5)   time = 2.053E-06, determ = 1.000E+20 Tau = 6.100E-08 3.021E-06
    6)   time = 1.527E-06, determ = 1.000E+20 Tau = 6.100E-08 3.021E-06
```

Figure 8. b

```
iter =  1, dumax = 6.82E-07, step = 1.00E+00
     Electron confinement time:  6.996E-06
    Wiffle radius rb:  2.692E-01
  787)   time = 1.636E-04, determ = 1.000E+20 Tau = 6.113E-08 2.911E-06
  788)   time = 1.663E-04, determ = 1.000E+20 Tau = 6.113E-08 2.911E-06
  789)   time = 1.676E-04, determ = 1.000E+20 Tau = 6.113E-08 2.911E-06
  790)   time = 1.736E-04, determ = 1.000E+20 Tau = 6.113E-08 2.911E-06
  791)   time = 1.744E-04, determ = 1.000E+20 Tau = 6.113E-08 2.911E-06
  792)   time = 1.690E-04, determ = 1.000E+20 Tau = 6.113E-08 2.911E-06
iter =  1, dumax = 4.24E-07, step = 1.00E+00
     Electron confinement time:  6.995E-06
    Wiffle radius rb:  2.692E-01
t(upscatter) = 0.000E+00 0.000E+00
Gun current 1 is:  1.800E+04
Gun current 2 is:  1.800E+04
Gun current 1 is:  1.800E+04
Gun current 2 is:  1.800E+04
Fusion Rate: DT=1.313E+14 DHe3=1.951E+10 DHe3 n=1.643E+11 pB11=5.434E+02
Gplasma: DD=6.624E-09 DT=1.417E-06 DHe3=9.550E-10 pB11=2.900E-18
Pbrem:  1.059E+01
Gicc= 9.130E-06 No/Nc = 9.523E+01 (<No>/Nc)^2= 1.378E+03

### SNAPSHOT SAVED ###

Restart file written:  brem2.tmp

Print files written:  brem2b.out
                      brem2b.tim
```

Figure 8. a. Beginning of the log file created by EKXL for the run in Figure 7.
b. End of the log file created by EKXL for the run in Figure 7.

```
Start date: 07/16/91    Start time: 17:29:20.81
Dataset:  brem2b
Time:  1.74E-04 sec
Number of Grid Points:  200
Deut Beam Neutron Count rate (#/sec):  DD=1.479E+12, DN=0.000E+00
                                                  Nb (#/m**3)
   R (m)       Pot (V)  DD (n/scm3)  Lambda (m)      Elec         Deut
0.000000, -1.453E+04,  1.718E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.000434, -1.453E+04,  1.718E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.000868, -1.453E+04,  1.718E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.001302, -1.453E+04,  1.719E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.001737, -1.453E+04,  1.720E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.002171, -1.453E+04,  1.722E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.002605, -1.453E+04,  1.724E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.003039, -1.453E+04,  1.727E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.003473, -1.453E+04,  1.731E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.003907, -1.453E+04,  1.736E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.004341, -1.453E+04,  1.744E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.004776, -1.453E+04,  1.753E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.005210, -1.453E+04,  1.765E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.005644, -1.453E+04,  1.780E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.006078, -1.453E+04,  1.799E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.006512, -1.453E+04,  1.821E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.006946, -1.453E+04,  1.845E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.007380, -1.453E+04,  1.870E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.007815, -1.453E+04,  1.893E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.008249, -1.453E+04,  1.908E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.008683, -1.453E+04,  1.909E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.009122, -1.453E+04,  1.885E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.009574, -1.453E+04,  1.820E+17,  0.000E+00,  6.601E+20,  6.601E+20
0.010045, -1.457E+04,  1.313E+17,  0.000E+00,  5.760E+20,  5.760E+20
0.010543, -1.464E+04,  7.805E+16,  0.000E+00,  4.414E+20,  4.414E+20
0.011077, -1.468E+04,  5.492E+16,  0.000E+00,  3.689E+20,  3.689E+20
0.011655, -1.472E+04,  4.027E+16,  0.000E+00,  3.150E+20,  3.150E+20
```

Figure 9. Beginning of the output dataset file generated by the EKXL run in Figure 7.

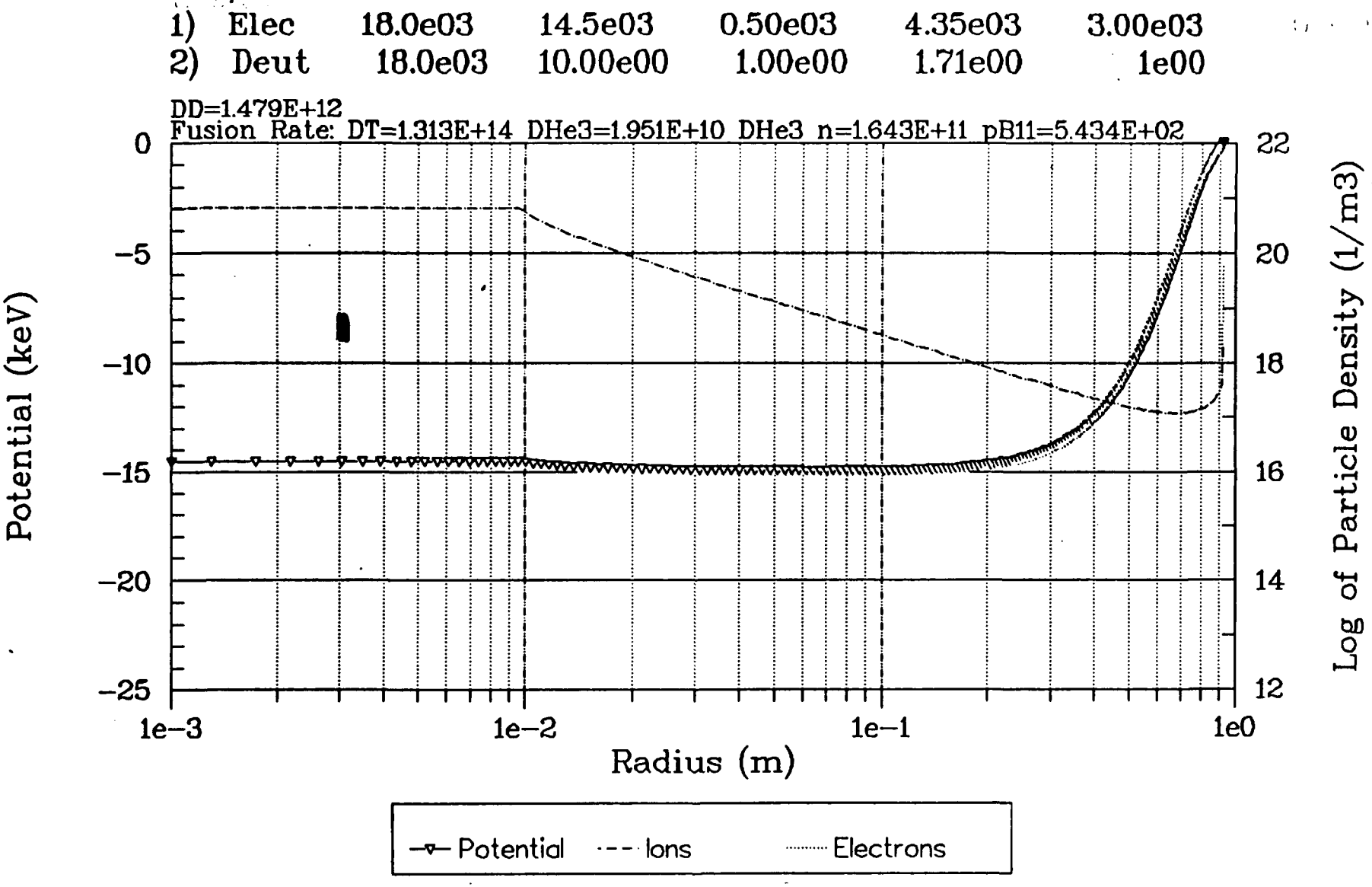

Figure 10 Sample plot of the data in the EKXL output file, as shown in Figure 9. This graph was generated in SuperCalc v5.0.